

**VŠB – Technická universita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Absolvování individuální odborné praxe**  
**Individual Professional Practise in the Company**

## Zadání bakalářské práce

Student: **Kamila Nykodýmová**  
Studijní program: B2647 Informační a komunikační technologie  
Studijní obor: 2612R059 Mobilní technologie  
Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practise in the Company**

### Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Městská nemocnice Ostrava, příspěvková organizace
2. Struktura závěrečné zprávy:
  - a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c. Zvolený postup řešení zadaných úkolů.
  - d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

### Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

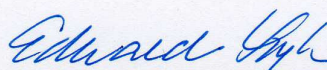
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Dr.Ing. Eduard Sojka**

Konzultant bakalářské práce: Ing. Vasil Waliszewski

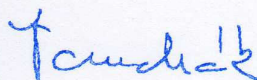
Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Dr.Ing. Eduard Sojka  
vedoucí katedry





prof. Ing. Ivo Vondrák, CSc.  
děkan fakulty

Prohlášení:

„Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.“

7. 5. 2010

Kamila Nykodýmová

### Poděkování:

Tímto děkuji Městské nemocnici Ostrava za umožnění absolvování bakalářské praxe. Rovněž děkuji vedoucímu bakalářské práce panu doc. Dr.Ing. Eduardu Sojkovi za odborné vedení a konzultantovi bakalářské práce panu Ing. Vasilu Waliszevskemu za poskytnuté konzultace, potřebné informace a věcné připomínky.

## **Abstrakt**

Bakalářská práce popisuje průběh mé bakalářské praxe v Městské nemocnici Ostrava. Její náplní bylo vytvoření nové verze aplikace pro oddělení klinické hematologie, protože původní už nevyhovovala současným požadavkům. Uživatelské rozhraní bylo zastaralé a aplikace nepodporovala některé, dnes nepostradatelné funkce, jako je například využití databázového serveru. V práci uvádím seznam zadaných úkolů, průběh jejich řešení a teoretické a praktické znalosti, které byly při jejich řešení uplatněny.

## **Abstract**

Bachelor's thesis describes the course of my bachelor's professional practice in the Municipal Hospital in Ostrava. Filling of professional practice was to create a new version of application for department of clinical hematology, because the original application hadn't suit the current requirements. The user interface was outdated and the application didn't support some of today indispensable function, such as the use of the database server. The work contains a list of assignments, course of their solutions and the theoretical and practical knowledge that has been applied to their solve.

## **Klíčová slova**

.NET, C#, Městská nemocnice Ostrava, Microsoft SQL Server 2005, Microsoft Visual Studio 2008, Panel, SQL dotazy, Uživatelské rozhraní, Windows Forms

## **Key words**

.NET, C#, Municipal Hospital Ostrava, Microsoft SQL Server 2005, Microsoft Visual Studio 2008, Panel, SQL queries, User interface, Windows Forms

## Obsah

|         |  |    |
|---------|--|----|
| 1.      | Úvod .....   | 1  |
| 2.      | Popis odborného zaměření firmy a zařazení studenta.....          | 2  |
| 3.      | Seznam úkolů a jejich časová náročnost.....                      | 2  |
| 3.1     | Vzhled aplikace .....  | 3  |
| 3.2     | Funkce aplikace .....  | 3  |
| 3.2.1   | Popis ponechaných funkcí .....                                   | 3  |
| 3.2.1.1 | Správa karet pacientů.....                                       | 3  |
| 3.2.1.2 | Kontroly .....   | 3  |
| 3.2.1.3 | Tisk .....   | 4  |
| 3.2.2   | Nové funkce aplikace.....  | 4  |
| 3.2.2.1 | Návrh databáze .....   | 4  |
| 3.2.2.2 | Správa číselníků.....  | 4  |
| 3.2.2.3 | Filtrování pacientů .....  | 4  |
| 3.2.2.4 | Vkládání obrazových příloh.....                                  | 5  |
| 4.      | Postup a řešení jednotlivých úkolů .....                         | 5  |
| 4.1.    | Tvorba celkového vzhledu aplikace.....                           | 5  |
| 4.2.    | Správa číselníků.....  | 7  |
| 4.3.    | Filtrování záznamů .....   | 8  |
| 4.4.    | Kontrola rodného čísla.....                                      | 8  |
| 5.      | Teoretické a praktické znalosti získané v průběhu studia.....    | 9  |
| 6.      | Znalosti či dovednosti scházející v průběhu odborné praxe .....  | 9  |
| 7.      | Dosažené výsledky v průběhu praxe a její celkové zhodnocení..... | 9  |
| 8.      | Literatura.....  | 10 |

## 1. Úvod

Tato bakalářská práce byla zpracována během praxe v Městské nemocnici Ostrava, na oddělení odboru informatiky pod vedením Ing. Vasilu Waliszewského. Cílem práce je vytvoření nové verze programu používaného na oddělení klinické hematologie k hodnocení cytologie kostní dřeně. Tento program dokumentuje výsledky laboratorního hodnocení kostní dřeně za účelem zjištění rakoviny krve. Protože zadaný úkol je značně rozsáhlý, je jeho zpracování rozděleno mezi dva studenty VŠB – TU fakulty elektrotechniky a informatiky.

Nejprve bylo nutné práci rozdělit na několik menších úkolů a domluvit harmonogram zpracování v závislosti na týmovém řešení uvedené problematiky.

Rozdělením a detailní specifikací těchto úkolů se zabývám v první části práce. Také v ní uvádím, které části aplikace budou vytvářeny mnou a které druhým členem týmu.

V další části je detailně popsána implementace jednotlivých úkolů, spolu s uvedením částí zdrojových kódů.

Poslední část je věnována znalostem získaných studií na VŠB-TUO, které jsem využila během praxe a jejímu závěrečnému shrnutí.

V následující kapitole je popsáno odborné zaměření firmy, ve které byla práce řešena.

## 2. Popis odborného zaměření firmy a zařazení studenta

Praxe byla vykonána v Městské nemocnici Ostrava, příspěvkové organizaci, konkrétně na Odboru informatiky. Vedoucím tohoto odboru je Ing. Vasil Waliszewski. V nemocnici se pracuje s produkty společnosti Microsoft. Z tohoto důvodu bude nová aplikace zpracována v programovacím jazyce C# a použitým databázovým serverem bude Microsoft SQL Server. Samotný vývoj aplikace bude probíhat ve vývojovém prostředí Microsoft Visual Studio 2008.

Následuje seznam všech úkolů a vymezení jejich časového zpracování.

## 3. Seznam úkolů a jejich časová náročnost

V tab. 1 je pro přehlednost uveden seznam jednotlivých úkolů s termínem jejich zpracování a zodpovědnou osobou za zpracování. Dále v textu jsou úkoly blíže popsány.

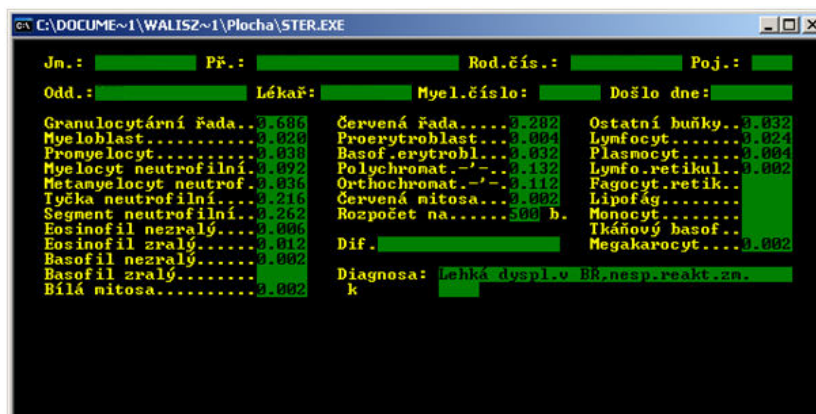
Tab. 1: Seznam úkolů

|     | Úkol                               | Termín zpracování | Zodpovědná osoba    |
|-----|------------------------------------|-------------------|---------------------|
| 1   | Nový vzhled aplikace               | Týden             | Nykodýmová          |
| 2   | Úprava stávajících funkcí aplikace | 3 měsíce          |                     |
| 2.1 | Správa karet pacientů              |                   | Nykodýmová          |
| 2.2 | Kontroly                           |                   | Nykodýmová, Glumbík |
| 2.3 | Tisk                               |                   | Glumbík             |
| 3   | Vývoj nových funkcí                | 2 měsíce          |                     |
| 3.1 | Návrh databáze                     |                   | Glumbík             |
| 3.2 | Správa číselníků                   |                   | Nykodýmová          |
| 3.3 | Filtrování pacientů                |                   | Nykodýmová          |
| 3.4 | Vkládání obrazových příloh         |                   | Glumbík             |



### 3.1 Vzhled aplikace

Prvním úkolem byla tvorba nového vzhledu aplikace pro hodnocení cytologie kostní dřeně. Dosavadní aplikace byla tvořena v programovacím jazyce Pascal, jako konzolová aplikace viz obr. 1. Základním nedostatkem v tomto programu bylo, že nešlo k navigaci využívat myš, jak je dnes většina uživatelů běžně zvyklá. Z tohoto důvodu by měla nová aplikace obsahovat moderní ovládací prvky a vytvořit tak uživateli co nejlepší pracovní prostředí.



Obr. 1 Původní vzhled aplikace

### 3.2 Funkce aplikace

Před tvorbou nové aplikace bylo nutné zvážit, které funkce z dosavadní aplikace musí zůstat zachované a které funkce bude třeba doplnit, aby aplikace splňovala všechny požadavky.

#### 3.2.1 Popis ponechaných funkcí

Z původní aplikace byly ponechány tři základní funkce. Správa karet pacientů, kontroly při tvorbě nové karty pacienta a tisk karty pacienta.

##### 3.2.1.1 Správa karet pacientů

Funkce správa karet pacientů umožňuje s kartami pacientů pracovat třemi způsoby. První možnost je tvorba nové karty pacienta. Při tvorbě nové karty vyplňuje uživatel jednoduchý formulář, obsahující základní údaje o pacientovi jako je rodné číslo, jméno, příjmení a pojišťovnu. Dále pak počet buněk získaný z pozorování vzorku kostní dřeně a diagnózu pacienta. Uživatel také může k pacientovi vložit komentář.

Další možností je vyhledávání karet pacientů. Požadovanou kartu je možné vyhledat pomocí rodného čísla, příjmení pacienta nebo čísla Myel, což je číslo, kterým v nemocnici označují kartu pacienta.

Poslední možností práce s kartami je jejich editace. Editace začíná vyhledáním karty pacienta. Po jejím vybrání se zobrazí editační formulář. Tento formulář má stejnou strukturu jako formulář při tvorbě nové karty pacienta. Jedinou změnou je, že všechny údaje jsou v něm už vyplněny.

##### 3.2.1.2 Kontroly

Kontroly jsou použity při vytváření nové karty pacienta. Rozlišujeme je na tvrdé a měkké. Při tvrdé kontrole je uživatel při pokusu o uložení karty upozorněn, že nevyplnil určitou položku a záznam není možné uložit do databáze, dokud tato položka nebude vyplněna. Při měkké kontrole

je uživatel upozorněn, že nevyplnil položku, ale na rozdíl od tvrdé kontroly je možné kartu uložit, pokud s tím uživatel souhlasí.

Speciální kontrola byla nastavena na rodné číslo. Pokud nebylo vloženo žádné číslo, byla použita tvrdá kontrola. Při vložení čísla, které neodpovídá parametrům rodného čísla v České republice, byla nastavena měkká kontrola. Důvod pro zvolení měkké kontroly byl, že ne každý pacient uvede své rodné číslo a také ne všechny rodná čísla cizinců odpovídají parametrům rodného čísla v ČR.

Tvrdé kontroly byly dále použity u příjmení pacientů a čísla Myel.

Měkké kontroly byly nastaveny pro kontrolní součet buněk. Buňky jsou rozděleny na granulocytární řadu, červenou řadu a ostatní buňky. Součet všech tří uvedených skupin musí dát dohromady 1000. Každá skupina má různé typy buněk, jejichž součet musí dát dohromady počet buněk jednotlivé řady.

### **3.2.1.3 Tisk**

Důležitá funkce aplikace je tisk. Uživatel si může nechat vytisknout detail karty pacienta. Výsledný dokument obsahuje všechny údaje z formuláře.

Následuje popis nových funkcí v aplikaci

## **3.2.2 Nové funkce aplikace**

Nové funkce byly navrženy pro jednodušší ovládání a manipulaci s aplikací. Místo ukládání dat o pacientech do souboru bylo zvoleno ukládání do databáze. Do databáze jsou také uloženy různé číselníky, ze kterých uživatel vybírá údaje při vytváření nebo editaci karty pacienta. Další novou funkcí je filtrování pacientů pro jejich rychlejší vyhledávání. Uživatelům je také jako novinka poskytnuta možnost vkládání obrazových příloh ke kartám pacientů.

Následuje detailní popis nových funkcí aplikace.

### **3.2.2.1 Návrh databáze**

Databáze je implementována v Microsoft SQL Serveru 2005 a byla navržena pomocí atributů z původní verze programu. K těm byly doplněny atributy určující cestu k obrazovým přílohám. Celá databáze je tvořena dvěma tabulkami a třemi číselníky.

První ze dvou tabulek má jako atributy všechny informace o pacientovi, získané hodnoty buněk a obrazové přílohy. Druhá tabulka se týká konfiguračního nastavení programu.

Tři číselníky obsahují informace, které se příliš často nemění. V případě změny je možné číselníky editovat, vytvářet nové položky nebo jednotlivé položky odstranit. První číselník obsahuje seznam pojišťoven České republiky, druhý seznam možných diagnóz pacientů a poslední číselník obsahuje oddělení, která jsou v nemocnici a ordinace externích lékařů.

### **3.2.2.2 Správa číselníků**

Správu číselníků je možné provádět v nastavení aplikace. Všechny již zmíněné číselníky lze editovat. Jednotlivé položky číselníků je také možné smazat, ale i vytvářet nové.

### **3.2.2.3 Filtrování pacientů**

Filtrace pacientů je použita při vyhledávání a editaci karet pacientů. Záznamy můžeme filtrovat podle třech atributů. Podle rodného čísla pacienta, podle čísla Myel nebo podle příjmení

pacienta. Filtrování probíhá hned při zadání prvního znaku. Výsledky vyhledávání jsou zobrazeny v ovládacím prvku DataGridView.

#### 3.2.2.4 Vkládání obrazových příloh

Uživateli je také umožněno vkládání obrazových příloh. Každý pacient má ve své kartě omezen počet příloh na pět. Cesta k příloze je uložena v databázi. Přílohy je možné ukládat i odstraňovat. Jednotlivé přílohy jsou zobrazeny vedle sebe ve zmenšeném měřítku. Pro zobrazení přílohy v jejím skutečném rozlišení je připraveno tlačítko Zobrazit. Pro vložení přílohy stačí myši kliknout na připravený ovládací prvek, který tím vyvolá dialogové okno pro vybrání přílohy.

## 4. Postup a řešení jednotlivých úkolů

Zadaný projekt byl rozdělen na jednotlivé etapy a zpracovatele jak je uvedeno v tab. 1. Tato bakalářská práce je tedy zaměřena na tvorbu celkového vzhledu aplikace, vytvoření funkčního formuláře pro tvorbu nové karty pacienta a formuláře pro editaci karty pacienta. Dalším úkolem byla správa číselníků, filtrování záznamů a kontrola rodného čísla.

### 4.1. Tvorba celkového vzhledu aplikace

Aplikace je tvořena jako Windows Forms Application. Základem je jedno centrální okno pro základní funkce aplikace. V programu je použito ještě jedno vedlejší okno, ve kterém může uživatel nastavovat číselníky a cesty k úložišti snímků.

Okno pro hlavní část programu je rozděleno na čtyři části: hlavičku aplikace, menu, střed aplikace a spodní informativní pruh viz obr. 2.



Obr. 2 Nová podoba aplikace

Barevně je celá aplikace sladěná do modrých barev podle internetových stránek nemocnice (<http://www.mnof.cz/>).

Hlavička programu obsahuje obrázky z hlavičky internetových stránek a název oddělení, kde bude program nasazen.

Menu aplikace je umístěné v levé části programu. Jednotlivé položky jsou seřazeny pod sebou, mají svůj název a jsou doplněny o obrázek napovídající, co která položka znamená. Texty jsou vloženy do ovládacího prvku Label a pro obrázky je použit ovládací prvek PictureBox. K zobrazení obsahu vytvoření nové karty, editaci karet a vyhledávání jednotlivých karet je použit ovládací prvek Panel. Tento prvek byl vybrán kvůli zamezení velkého množství otevřených oken, které by vznikaly při použití ovládacího prvku Form. Ovládací prvek Form je použit pouze pro zobrazení nastavení aplikace a pro zobrazení výběru tiskárny.

Ve Formu pro nastavení jsou použity následující ovládací prvky. Prvek Label pro popisné informace, k vložení textu uživatelem je použit ovládací prvek TextBox, pro výběr z nabídky je použit prvek ComboBox a posledním prvkem je prvek Button pro vyvolání zvolené události.

Pro vytvoření nové karty pacienta je vytvořen Panel, ve kterém jsou ovládací prvky pro vložení záznamů o pacientovi. Formulář je tvořen ovládacím prvkem Label pro označení údajů, které má uživatel vyplnit. K vložení údajů o pacientovi je použit prvek TextBox. Pro vložení dne, kdy došlo k vytvoření nové karty je použit ovládací prvek DateTimePicker. Uživatel si jednoduše zvolí požadované datum a nedojde tak k vložení špatného formátu datumu. Pro výběr údajů z číselníků je zvolen ComboBox. Vložit komentář je možné v ovládacím prvkem TextBox s nastaveným parametrem Multiline na true. To umožňuje vložení více řádků textu. Karta pacienta se uloží do databáze po stisku tlačítka Uložit kartu, kdy je volána metoda obsluhující událost kliknutí na toto tlačítko. V této metodě se převádí zadané hodnoty pro kontrolní součet na typ double a probíhají kontroly. Pokud kontroly proběhnou správně je volána metoda ulozitKartu(). Tato metoda uloží kartu pacienta do databáze pomocí SQL dotazu.

Pro editaci karet jsou použity dva Panely. V prvním Panelu dochází k vyhledání požadovaného pacienta. Panel obsahuje ovládací prvek DataGridView. Tento prvek zobrazuje všechny pacienty uložené v databázi s parametry id, rodné číslo, číslo myel, jméno pacienta, příjmení pacienta a jméno lékaře. Pro naplnění tohoto prvku po filtraci je použita metoda fillDataGridKarty(). V této metodě se musí nejdříve vytvořit připojení k databázi pomocí instance třídy SqlConnection s parametrem, který obsahuje connectionString a zavoláním metody třídy SqlConnection Open(). Po připojení je vytvořen objekt SqlDataAdapter, který přebírá dva parametry. Jeden parametr tvoří SQL dotaz pro výběr informací z databáze a druhý parametr pro připojení k požadované databázi. Následující část kódu zobrazuje připojení k databázi a SQL dotaz:

```
connection = new SqlConnection(connectionString);
connection.Open();
SqlDataAdapter a = new SqlDataAdapter("SELECT id, rc, myel, jmeno, prijmeni, lekar FROM
[NemocniceOstrava].[dbo].[pacienti]", connection);
```

Následuje uložení výsledků vyhledávání z databáze. Pro uložení je použita instance třídy DataTable a metoda fill() třídy SqlDataAdapter.

Pro jednodušší vyhledávání je použita filtrace podle rodného čísla, čísla Myel a příjmení pacienta. Tyto atributy uživatel vkládá do ovládacího prvku TextBox. Pro vybrání konkrétního pacienta stačí myši poklepat na řádek s údaji o pacientovi. Tato akce zobrazí nový Panel.

V tomto panelu je stejný formulář jako při tvorbě nové karty pacienta. Rozdíl je, že údaje vyplněné při vytváření nové karty jsou již předvyplněny.

Pro vyhledávání karet pacienta jsou použity také dva Panely. V prvním panelu probíhá vyhledávání stejně jako u editace karet. Opět je použit ovládací prvek DataGridView pro zobrazení pacientů z databáze. Také je zde použita filtrace pro jednodušší vyhledání záznamu o pacientovi.

Druhý Panel se zobrazí, pokud uživatel vybere konkrétní řádek s údaji o pacientovi, kterého vyhledával. V tomto panelu jsou pak zobrazeny detaily o pacientovi, které se ovšem nedají editovat. K úpravě údajů slouží tlačítko Editovat, které uživatele přesměruje do editční části programu. Také se na panelu nachází tlačítko k zobrazení příloh, které karta pacienta obsahuje. Posledním tlačítkem je tlačítko volající metodu tisku. Po aktivování tohoto tlačítka se uživateli nabídne výběr tiskáren nainstalovaných na jeho počítači.

Ve spodní části aplikace se nachází informativní pruh, vypisující uživateli důležité zprávy. V závislosti na typu informace se pruh zbarvuje buď do červena nebo do zelena.

## 4.2. Správa číselníků

Aplikace obsahuje tři číselníky. První číselník obsahuje seznam oddělení, která jsou v nemocnici. Kromě nich jsou, ale v číselníku uvedena také externí pracoviště. Druhý číselník je seznam pojišťoven v České republice a třetí obsahuje seznam lékařských diagnóz.

Samotná správa probíhá v okně s nastavením aplikace. Uživatel zde může přidávat, editovat a mazat položky zvoleného číselníku. Při vytváření nové položky se do proměnné `id_ciselniku` uloží číslo identifikující, který číselník byl pro přidání zvolen. Dále se otevře nové formulářové okno pro vyplnění informací potřebných ke správnému přidání položky do číselníku. Pro editaci se využívá stejného formulářového okna jen s tím rozdílem, že údaje jsou již vyplněny. Údaje se do číselníku (databáze) ukládají pomocí jednoduchého SQL dotazu uvedeného níže:

```
string query = "INSERT INTO [NemocniceOstrava].[dbo].[pojistovny] ( [cislo], [nazev]) VALUES ('" +  
textBox1.Text + "', '" + textBox2.Text + "')";
```

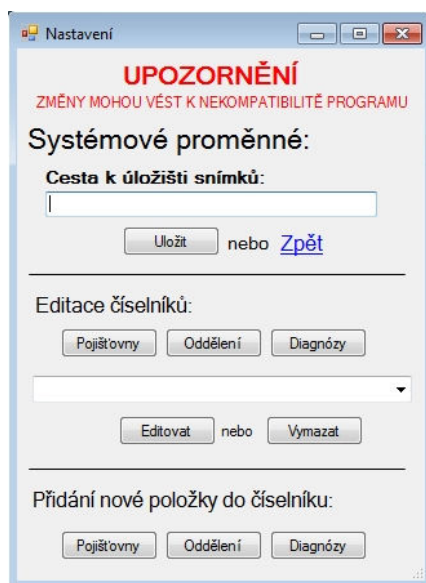
Tento dotaz vkládá nový záznam pojišťovny do databáze. Obdobný dotaz je použit i při vytváření nové diagnózy nebo při tvorbě nového oddělení.

Pro editaci číselníků musí uživatel zvolit, který číselník chce editovat. Tento výběr se projeví v ComboBoxu, který se tak naplní daty z databáze.

Poslední možností jak spravovat číselníky je mazání jednotlivých položek z číselníků. V ComboBoxu si uživatel vybere požadovanou položku a po zmáčknutí tlačítka Vymazat dojde k odstranění položky z databáze. Ukázka SQL dotazu pro odstranění položky z číselníku pojišťoven:

```
SqlCommand command = new SqlCommand("DELETE FROM  
[NemocniceOstrava].[dbo].[pojistovny] WHERE id='" + id + "'", connection);
```

Po úspěšném vymazání položky se v ComboBoxu zobrazí zpráva: Položka byla úspěšně vymazána.



Obr. 3 Formulář pro práci s číselníky

### 4.3. Filtrování záznamů

Filtrování záznamů se používá při vyhledávání pacienta. Filtry jsou nastaveny na rodné číslo, číslo Myel a příjmení pacienta. Když uživatel začne zadávat například rodné číslo, je vyvolávána událost `OnTextChanged`, na kterou pomocí přiřazené metody reagujeme tak, že provádíme v prvku `DataGriedView` novým dotazem výběr položek odpovídajících našim zadaným požadavkům. Pokud uživatel z nějakého důvodu nenašel požadovaný záznam, může k vyhledání zvolit jiný filtr. Pro vyhledávání je použito upraveného SQL dotazu. Vyhledanými záznamy naplníme objekt `DataTable`, který můžeme použít jako zdroj dat (`DataSource`) pro `DataGridView`. Ukázka SQL dotazu pro vyhledávání položek dle rodného čísla pacienta:

```
SqlDataAdapter a = new SqlDataAdapter ("SELECT id, rc, jmeno, prijmeni, myel FROM [NemocniceOstrava].[dbo].[pacienti] WHERE rc LIKE '" + dotaz + "%'", connection)
```

### 4.4. Kontrola rodného čísla

Rodné číslo je jednoznačný číselný identifikátor člověka. Lze z něj vyčíst datum narození a pohlaví příslušné osoby. V aplikaci je kontrola rodného čísla aplikována jako měkká kontrola.

Rodná čísla do roku 1954 měla za lomítkem jen tři číslice. Po tomto roce přibyla čtvrtá číslice, sloužící jako kontrolní číslice. Čtvrté číslo se doplňuje tak, aby celé rodné číslo bylo beze zbytku dělitelné jedenácti.

Jako první při kontrole určíme, kolik má vyplněné rodné číslo číslic. Má-li jiný počet nežli 9 nebo 10 jedná se o chybně zadané číslo. Následně u devíti místného čísla kontrolujeme prvních šest číslic, které nám udávají datum narození ve formátu `RRMMDD`. Pro `RR` jsou povolenými hodnotami 00 až 53, pro `MM` je to 01 až 12 pro muže a 51-62 pro ženy. Pro `DD` je to v závislosti

na MM 01 až 31. U desíti místných rodných čísel kromě toho navíc kontrolujeme i celkovou dělitelnost rodného čísla číslem 11.

## **5. Teoretické a praktické znalosti získané v průběhu studia**

Při vytváření aplikace jsme specifikovali požadavky na funkčnost aplikace a navrhli vhodné postupy pro tvorbu projektu. Tyto postupy jsem se naučila v předmětu Úvod do softwarového inženýrství.

Pro tvorbu databáze a práci s ní jsem čerpala z předmětů Teorie zpracování dat a Správa databází, ve kterých jsem se také naučila vytvářet SQL dotazy.

Při programování jsem zvolila objektové orientovaný přístup, jehož základy jsem získala v předmětech Úvod to programování (Java) a Programovací techniky.

## **6. Znalosti či dovednosti scházející v průběhu odborné praxe**

Celá aplikace je tvořena v jazyce C#, se kterým jsem se bohužel v průběhu studia nesetkala, ale před započítím praxe jsem se s ním dostatečně seznámila.

## **7. Dosažené výsledky v průběhu praxe a její celkové zhodnocení**

Cílem bakalářské práce bylo vytvoření nové verze stávající aplikace pro oddělení klinické hematologie.

Nejdůležitějšími body práce byla změna vzhledu aplikace a změna způsobu ukládání dat. Konzolová aplikace byla změněna na Windows Forms Application, aby bylo možné použít moderních ovládacích prvků. K ukládání dat byl využit databázový server.

Během průběhu praxe byly výsledky zpracování po jednotlivých částech konzultovány v nemocnici na oddělení odboru informatiky. V závěru praxe byla aplikace nainstalována v nemocnici a otestována na oddělení klinické hematologie. Tím byl splněn cíl této praxe. Při testování jsme obdrželi drobné připomínky a návrhy na úpravy v aplikaci. Tyto úpravy budou probíhat, i po ukončení bakalářské praxe.

Přínosem pro mě bylo nejen zlepšení svých znalostí v programovacím jazyce C#, ale také zkušenost s prací v týmu.

## 8. Literatura

NAGEL, Christian a kolektiv. C# 2005 Programujeme profesionálně. Nakladatelství Computer Press, 2007. 1400 s. ISBN: 80-251-1181-4.

KENT, Jeff. Visual C# 2005 bez předchozích znalostí: průvodce pro samouky. Nakladatelství Computer Press, 2007. 307 s. ISBN: 978-80-251-1584-8.

PETZOLD, Charles. Programování Microsoft Windows Forms v jazyce C#. Nakladatelství Computer Press, 2006. 351 s. ISBN: 80-251-1058-3